



# Adaptive Algorithms for Sparse Nonlinear Channel Estimation

## Citation

Kalouptsidis, Nicholas, Gerasimos Mileounis, Behtash Babadi, and Vahid Tarokh. 2009. Adaptive algorithms for sparse nonlinear channel estimation. Paper presented at the 2009 IEEE Workshop on Statistical Signal Processing, Cardiff, Wales, UK.

## Permanent link

<http://nrs.harvard.edu/urn-3:HUL.InstRepos:4481494>

## Terms of Use

This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Open Access Policy Articles, as set forth at <http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#OAP>

## Share Your Story

The Harvard community has made this article openly available.  
Please share how this access benefits you. [Submit a story](#).

[Accessibility](#)

# ADAPTIVE ALGORITHMS FOR SPARSE NONLINEAR CHANNEL ESTIMATION

Nicholas Kalouptsidis<sup>#</sup>, Gerasimos Mileounis<sup>#</sup>, Behtash Babadi<sup>\*</sup> and Vahid Tarokh<sup>\*</sup>

<sup>#</sup>*Dept. of Informatics and Telecommunications, University of Athens, Ilisia 15784, Greece*

<sup>\*</sup>*School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138*

*Email: {kalou, gmil}@di.uoa.gr, {behtash, vahid}@seas.harvard.edu*

## ABSTRACT

In this paper, we consider the estimation of sparse nonlinear communication channels. Transmission over the channels is represented by sparse Volterra models that incorporate the effect of Power Amplifiers. Channel estimation is performed by compressive sensing methods. Efficient algorithms are proposed based on Kalman filtering and Expectation Maximization. Simulation studies confirm that the proposed algorithms achieve significant performance gains in comparison to the conventional non-sparse methods.

**Index Terms**— Volterra series, Adaptive estimation, Compressive sensing, Kalman filtering, Expectation Maximization.

## I. INTRODUCTION

Channel nonlinearities are mainly due to Power Amplifiers (PA). PAs located at an access point of a downlink channel (base stations in cellular systems and repeaters for satellite links) often operate close to saturation in order to achieve power efficiency. The models employed in the description of PAs are either static (memoryless) or dynamic (models with memory).

Wireless communication channels are characterized by time varying multipath propagation effects. Quite often in practice, several reflections reach the receiver at different time instances. These reflections arrive at the receiver with longer delay than the first group. Hence, the wireless channel is modeled by sparse fading rays and long zero samples and thus admits a sparse representation [1]. The sparseness characteristic is preserved when the PA representation is also described by a sparse model [2]. Recent experimental results reported in [2] indicate better performance if sparse nonlinear models are employed for the representation of PA. Moreover, the time-varying nature of the wireless channels suggest the use of adaptive algorithms that minimize transmission delays and take advantage of parameters sparsity. Thus, compressive sensing provides a promising framework for such developments. Adaptive algorithms for sparse channel estimation are developed in [3, 4]. In [3] two different sparsity constraints are incorporated into the quadratic cost function of the LMS algorithm, to take into account the sparse channel coefficient vector. An  $\ell_1$ -regularized RLS type algorithm based on a low complexity Expectation-Maximization, is derived in [4].

In this paper we focus on adaptive estimation of sparse nonlinear communication channels. Adaptation is carried out

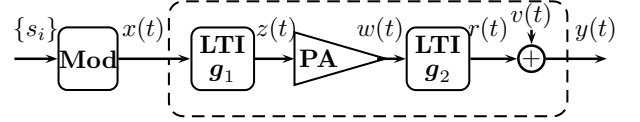


Fig. 1. Digital satellite link

by recursive algorithms that combine Expectation Maximization and Kalman filtering. The expectation step is carried out by Kalman filtering while the maximization step corresponds to a soft-thresholding function due to the  $\ell_1$  regularization.

The rest of the paper is organized as follows. Nonlinear channels and sparse channel estimation are discussed in Section II. The proposed algorithms for adaptive tracking of sparse nonlinear channels are given in section III. Simulation results are presented in Section IV. Conclusions are discussed in Section V.

## II. SPARSE NONLINEAR CHANNEL ESTIMATION

In what follows, power amplifier nonlinear models are incorporated into the study of two important channels: 1) the satellite link, and 2) the multi-path wireless channel. In both cases, the overall communication channel is represented by baseband Volterra series.

### A. Nonlinear channel models

In satellite digital transmission, both the earth station and the satellite repeater employ power amplifiers. The satellite amplifier operates near saturation due to limited power resources and hence behaves in a nonlinear fashion. The satellite link is represented by the block diagram of Fig. 1. The LTI filter with impulse response  $g_1$  describes the cascade of all linear operations preceding the power amplifier. Likewise the LTI filter  $g_2$  represents the cascade of all linear devices following the nonlinearity.

An analysis of the above system for static power amplifiers is provided by Benedetto and Biglieri [5]. Let us next consider power amplifiers with memory described by Volterra models. To reduce the computational complexity we shall follow standard practice [2, 6] and confine our study to diagonal Volterra models<sup>1</sup>. Straightforward calculation lead to the

<sup>1</sup>The analysis of general (non-diagonal) Volterra models is similar.

baseband Volterra model

$$r(t) = \sum_{p=0}^{\lfloor \frac{P-1}{2} \rfloor} \int \cdots \int h_{2p+1}(\tau_{1:2p+1}) \times \prod_{i=1}^{p+1} x(t - \tau_i) \prod_{j=p+2}^{2p+1} x^*(t - \tau_j) d\tau_{1:2p+1}. \quad (1)$$

where  $h_{2p+1}(\tau_{1:2p+1})$  denotes the baseband kernel with  $\tau_{1:2p+1} = (\tau_1, \dots, \tau_{2p+1})$ .

In most cases the filter  $g_1$  performs a specific functionality (for instance pulse shaping) and hence is known. Since in this paper we shall deal with channel estimation using known inputs, we may with no loss of generality assume the input signal is the output of  $g_1$ . In this case the Volterra representation from signal  $x(t)$  to signal  $r(t)$  gets simpler. More precisely we have

$$r(t) = \sum_{p=0}^{\lfloor \frac{P-1}{2} \rfloor} \int h_{2p+1}(t, \tau_1) |x(t - \tau_1)|^{2p} x(t - \tau_1) d\tau_1. \quad (2)$$

The above expression represents also the multipath channel. In this case the modulated signal is amplified by a power amplifier and then transmitted through the wireless medium. The received waveform is the superposition of weighted and delayed versions of the signal resulting from various multipaths plus additive white Gaussian noise. We shall assume that the different nonzero fading rays arrive at the receiver at different time instances and they vary slowly with time and frequency hence the wireless channel becomes a frequency selective channel [1] and is described by an impulse response of the form

$$g_2(\rho) = \sum_{i=1}^N a_i \delta(\rho - \tau_i) \quad (3)$$

where  $N$  is the number of paths,  $a_i$  is the attenuation along path  $i$  and  $\tau_i$  is the clustered delay.

## B. Sparse channel estimation

The transmission systems described in the previous section operate in continuous time. Discrete Volterra forms result when the modulation at the transmitter and the sampling device at the receiver are taken into account. We shall consider memoryless modulation schemes whereby

$$x(t) = \sum_i s_i \delta(t - iT_s). \quad (4)$$

The sequence  $s_i$  consists of i.i.d (discrete) complex valued random variables and  $T_s$  denotes the symbol period. Substituting  $x(t)$  from Eq. (4) into (1) yields the discrete baseband Volterra model [5] which can be expressed as a linear regression. Let us define the vector

$$\mathbf{x}_{i, M_{2p+1}} = [s_i, s_{i-1}, \dots, s_{i-M_{2p+1}}]^T$$

and the  $i$ -fold Kronecker product

$$\mathbf{x}_{i, M_{2p+1}}^{(p+1, p)} = [\otimes_{j=1}^{p+1} \mathbf{x}_{i, M_{2p+1}}] \otimes [\otimes_{k=1}^p \mathbf{x}_{i, M_{2p+1}}^*].$$

The Kronecker product contains all  $2p + 1$  order products of the input with  $p$  conjugate copies. The output can be written in the linear regression form

$$y_i = [\mathbf{x}_{i, M_1}^T \cdots \mathbf{x}_{i, M_{2p+1}}^{(p+1, p)T}] \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_{2p+1} \end{bmatrix} + v_i \quad (5)$$

with  $1 \leq i \leq n$ . If we stack  $n$  successive samples in a column format we obtain

$$\mathbf{y}_n = \mathbf{X}_n \mathbf{h} + \mathbf{v}_n \quad (6)$$

where  $\mathbf{y}_n = [y_1, \dots, y_n]^T$ ,  $\mathbf{v}_n = [v_1, \dots, v_n]^T$  and  $\mathbf{X}_n = [\mathbf{x}_1^T, \dots, \mathbf{x}_n^T]^T$ .

Eq. (6) provides a noisy representation of a block of received successive samples in terms of the columns of  $\mathbf{X}_n$  (also referred to as dictionary), that are formed by the products of shifted symbol sequences. The above representation is sparse and hence recovery of the vector  $\mathbf{h}$  can be accomplished by compressed sensing methods. Next we consider sparsity. It is well documented in the literature that parsimonious models are highly desirable in the representation of memory PA. In fact it has been experimentally observed [2] that sparse diagonal Volterra models provide enhanced performance in comparison to the full model. Furthermore, a physical justification of sparsity for the multipath channel is given in [1]. The sparsity of the  $2p + 1$  kernel is at most  $s_k \times s_m$ , where  $s_k$  is the sparsity of the PA and  $s_m$  is the sparsity of the multipath coefficients. Similar observations hold for the satellite channel. It thus follows that the vector  $\mathbf{h}$  in Eq. (6) is sparse.

Recovery of the locations, the magnitudes and the non-linear coefficients of  $\mathbf{h}$  can be accomplished by the convex program

$$\min_{\mathbf{h}} \left\{ \frac{1}{2} \|\mathbf{y}_n - \mathbf{X}_n \mathbf{h}\|_{\ell_2}^2 + \gamma \|\mathbf{h}\|_{\ell_1} \right\}. \quad (7)$$

The  $\ell_1$ -norm provides a convex relaxation to the  $\ell_0$  quasi-norm. The scalar parameter  $\gamma$  provides a trade-off between sparsity and total squared error. The optimization problem (7) has been widely studied from the perspective of compressive sensing (see, for instance, [7]).

## III. ADAPTIVE ALGORITHMS

Since the parameter vector  $\mathbf{h}$  changes with time we need a model that captures the corresponding dynamics. A popular technique in the adaptive filtering literature is to describe parameter variation by the first-order model [8]

$$\mathbf{h}_n = \mathbf{h}_{n-1} + \mathbf{q}_{n, \Lambda_0} = \mathbf{h}_0 + \sum_{i=1}^n \mathbf{q}_{i, \Lambda_0}; \quad \mathbf{h}_0 \sim \mathcal{N}(\bar{\mathbf{h}}_0, \sigma_0^2 \mathbf{I}_{\Lambda_0}). \quad (8)$$

**Table 1.** Algorithms for sparse nonlinear channel identification

<p><b>(a) EM-Kalman</b></p> <hr/> <p><b>Initialization :</b> <math>\mathbf{h}_0 = \bar{\mathbf{h}}_0</math>, <math>\mathbf{P}_0 = \delta^{-1} \mathbf{I}</math> with <math>\delta = \text{const.}</math></p> <p><b>For</b> <math>n := 1, 2, \dots</math> <b>do</b></p> <p>1: <math>\mathbf{k}_n = \frac{\mathbf{P}_{n-1} \mathbf{x}_n^*}{\sigma_v^2 + \mathbf{x}_n^T \mathbf{P}_{n-1} \mathbf{x}_n^*}</math></p> <p>2: <math>\boldsymbol{\psi}_n = \mathbf{h}_{n-1} + \mathbf{k}_n \varepsilon_n</math></p> <p>3: <math>\mathbf{P}_n = \mathbf{P}_{n-1} - \mathbf{k}_n \mathbf{x}_n^T \mathbf{P}_{n-1} + r_{n-1} \mathbf{I}</math></p> <p>4: <math>\sigma_{\varepsilon_n}^2 = \sigma_v^2 + \mathbf{x}_n^T \mathbf{P}_{n-1} \mathbf{x}_n^*</math></p> <p>5: <math>r_n = \alpha r_{n-1} + (1 - \alpha) R \left( \frac{ \varepsilon_n ^2 - \sigma_{\varepsilon_n}^2}{\mathbf{x}_n^T \mathbf{x}_n^*} \right)</math></p> <p>6: <math>\sigma_v^2 = \alpha \sigma_v^2 + (1 - \alpha) R ( \varepsilon_n ^2 - \mathbf{x}_n^T \mathbf{P}_{n-1} \mathbf{x}_n^*)</math></p> <p>7: <math>\mathbf{h}_n = \text{sgn}(\boldsymbol{\psi}_n) \left[  \boldsymbol{\psi}_n  - \gamma (\sigma_0^2 + r_n) \mathbf{I} \right]_+</math></p> <p><b>end For</b></p>	<p><b>(b) EM-RLS</b></p> <hr/> <p><b>Initialization :</b> <math>\mathbf{h}_0 = 0</math>, <math>\mathbf{P}_0 = \delta^{-1} \mathbf{I}</math> with <math>\delta = \text{const.}</math></p> <p><b>For</b> <math>n := 1, 2, \dots</math> <b>do</b></p> <p>1: <math>\mathbf{k}_n = \frac{\mathbf{P}_{n-1} \mathbf{x}_n^*}{\lambda + \mathbf{x}_n^T \mathbf{P}_{n-1} \mathbf{x}_n^*}</math></p> <p>2: <math>\boldsymbol{\psi}_n = \mathbf{h}_{n-1} + \mathbf{k}_n \varepsilon_n</math></p> <p>3: <math>\mathbf{P}_n = \lambda^{-1} \mathbf{P}_{n-1} - \lambda^{-1} \mathbf{k}_n \mathbf{x}_n^T \mathbf{P}_{n-1}</math></p> <p>4: <math>\mathbf{R}_n = (1 - \lambda) \mathbf{P}_n</math></p> <p>5: <math>\mathbf{h}_n = \text{sgn}(\boldsymbol{\psi}_n) [ \boldsymbol{\psi}_n  - \gamma (\sigma_0^2 + \text{diag}(\mathbf{R}_n))]_+</math></p> <p><b>end For</b></p> <hr/> <p><b>(c) EM-LMS</b></p> <hr/> <p><b>Initialization :</b> <math>\mathbf{h}_0 = 0</math>, <math>0 &lt; \mu &lt; 2\lambda_{\max}^{-1}</math></p> <p><b>For</b> <math>n := 1, 2, \dots</math> <b>do</b></p> <p>1a: <math>\mathbf{h}_n = \mathbf{h}_{n-1} - \gamma \text{sgn}(\mathbf{h}_{n-1}) + \mu \mathbf{x}_n^* \varepsilon_n</math></p> <p>1b: <math>\mathbf{h}_n = \mathbf{h}_{n-1} - \gamma \frac{\text{sgn}(\mathbf{h}_{n-1})}{1 + \epsilon \ \mathbf{h}_{n-1}\ } + \mu \mathbf{x}_n^* \varepsilon_n</math></p> <p><b>end For</b></p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

$$\text{sgn}(z) = z/|z| \text{ for any } z \in \mathbb{C} \text{ except } z = 0$$

$$(z)_+ = \max(\Re(z), 0) + j \max(\Im(z), 0)$$

$\Lambda_0$  denotes the support set of  $\mathbf{h}_0$ , i.e. the set of the non-zero coefficients. The noise term  $\mathbf{q}_n$  is zero outside  $\Lambda_0$  and zero-mean Gaussian inside  $\Lambda_0$  with diagonal covariance matrix  $\mathbf{R}_{n, \Lambda_0} = \text{diag}[\sigma_{q_1}^2(n), \dots, \sigma_{q_d}^2(n)]$ , where  $d$  is the  $\ell_0$  norm of  $\mathbf{h}_0$ . The variances  $\{\sigma_{q_i}^2(n)\}_{i=1}^d$  are in general allowed to vary with time. The stochastic processes  $\mathbf{v}$ ,  $\mathbf{q}$  and the random variable  $\bar{\mathbf{h}}_0$  are mutually independent.

We next incorporate Eq. (8) and the convex program (7) into the Expectation-Maximization (EM) framework. The resulting adaptive algorithms employ only one iteration per time update for computational purposes. Let  $\boldsymbol{\theta} = \bar{\mathbf{h}}_0$  be the vector of unknown parameters. Note that under the Gaussian assumption postulated above, minimization of (8) is equivalent to the maximization of the log-likelihood  $p(\mathbf{y}_n | \boldsymbol{\theta})$  augmented by an  $\ell_1$  penalty.

To apply the Expectation-Maximization method we have to specify the complete and incomplete data. The vector  $\mathbf{h}_n$  at time  $n$  is taken to represent the complete data vector, whereas  $\mathbf{y}_{n-1}$  accounts for the incomplete data [9]. In this context the conditional density  $p(\mathbf{h}_n | \mathbf{y}_{n-1})$  plays a major role. This density is Gaussian with mean and covariance:

$$\boldsymbol{\psi}_n = \mathbb{E}[\mathbf{h}_n | \mathbf{y}_{n-1}]$$

$$\mathbf{P}_n = \mathbb{E}[(\mathbf{h}_n - \boldsymbol{\psi}_n)(\mathbf{h}_n - \boldsymbol{\psi}_n)^H] = \sigma_0^2 \mathbf{I} + \sum_{t=1}^n \text{diag}[\sigma_{q_i}^2(t)].$$

Under broad conditions the maximizer of the incomplete likelihood is obtained by maximizing the complete likelihood function through successive application of the following two steps:

**E-step :** computes the conditional expectation

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_{n-1}) = \mathbb{E}_{p(\mathbf{h}_n | \mathbf{y}_{n-1}, \hat{\boldsymbol{\theta}}_{n-1})} [\log p(\mathbf{h}_n; \boldsymbol{\theta})] \quad (9)$$

**M-step :** maximizes the  $Q$ -function minus the  $\ell_1$ -penalty with respect to  $\boldsymbol{\theta}$ :

$$\hat{\boldsymbol{\theta}}_n = \arg \max_{\boldsymbol{\theta}} \left\{ Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_{n-1}) - \gamma \|\boldsymbol{\theta}\|_{\ell_1} \right\}. \quad (10)$$

Note that

$$\log p(\mathbf{h}_n; \boldsymbol{\theta}) = \text{const.} - \frac{1}{2} (\mathbf{h}_n - \boldsymbol{\psi}(\boldsymbol{\theta}))^H \mathbf{P}_n^{-1} (\mathbf{h}_n - \boldsymbol{\psi}(\boldsymbol{\theta})).$$

Therefore the  $Q$ -function takes the form

$$Q(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}_{n-1}) = \text{const.} + \boldsymbol{\theta}^H \mathbf{P}_n^{-1} \boldsymbol{\psi}_n - \frac{1}{2} \boldsymbol{\theta}^H \mathbf{P}_n^{-1} \boldsymbol{\theta} \quad (11)$$

where the constant incorporates all terms that do not involve  $\boldsymbol{\theta}$  and hence do not affect the maximization.

The parameter  $\boldsymbol{\psi}_n$  is recursively computed by the Kalman filter [8], see Table 1(a) steps 1 – 3, which in the special case of the time-varying random walk model Eq. (8) takes an RLS type appearance. Note that  $\varepsilon_n$ , in Table 1, denotes the prediction error given by  $\varepsilon_n = y_n - \mathbf{x}_n^T \mathbf{h}_{n-1}$ .

Maximization of the  $Q$  function leads to the *soft thresholding* function

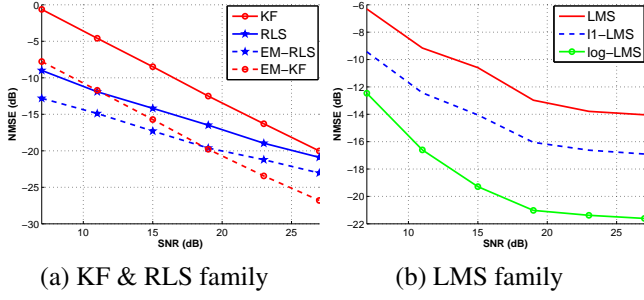
$$\hat{\mathbf{h}}_{n,i} = \text{sgn}(\boldsymbol{\psi}_{n,i}) \left[ |\boldsymbol{\psi}_{n,i}| - \gamma \left( \sigma_0^2 + \sum_{t=1}^n \sigma_{q_i}^2(t) \right) \right]_+ \quad (12)$$

This operation shrinks coefficients above the threshold in magnitude value.

**EM-Kalman filter.** The Kalman filter computes  $\hat{\mathbf{h}}_n$  under the assumption that the variances  $\sigma_v^2$  and  $\{\sigma_{q_i}^2(n)\}_i^d$  are known. The noise variances can be estimated in various ways. One method is to use the Maximum Likelihood estimates. These estimates can be obtained by maximizing the  $Q$ -function.

Alternatively, under the assumption that the state noise is  $\mathbf{R}_{n, \Lambda_0} = r_n \mathbf{I}$ , then both noise disturbances can be estimated adaptively. A smoothed estimate of the state and observation noise can be respectively obtained according to steps 5 and 6 of Table 1(a), where  $\alpha$  is a smoothing parameter and  $R(x)$  is the ramp function ( $R(x) = x$  if  $x \geq 0$  and 0 otherwise). These two methods for online estimation of the noise disturbances is due to Jazwinski [10].

**EM-RLS filter.** The recursive procedure for the determination of the Kalman filter in the case of the random-walk



**Fig. 2.** NMSE of the three sparse adaptive algorithms

model, Eq. (8), resembles the RLS algorithm. In fact, the RLS can be viewed as a special form of Table 1(a) which provides an alternative for the estimation of the noise variances.

The RLS filter is given by steps 1-3 of Table 1(b) [8], with  $\sigma_v^2 = \lambda$ ,  $R_n = (1 - \lambda)P_n$ .

**Sparse LMS filter.** For the purposes of simulations presented in the next section we discuss the LMS variant developed in [3]. LMS updates some convex cost function of the prediction error signal  $\varepsilon_n$  plus an  $\ell_1$  penalty. The update equation which minimizes the cost function is given in step 1a of Table 1(c). The authors in [3] replace the  $\ell_1$ -norm penalty by the log-sum penalty function. Hence, the resulting update equation for this cost function becomes step 1b of Table 1(c). The log-sum penalty function has the potential of being more-sparsity encouraging since it better approximates the non-convex  $\ell_0$ -norm.

#### IV. SIMULATIONS

Experiments were conducted on the multipath channel setup of Eq. (2). The algorithms were run for 2000 iterations and averaged over 50 Monte Carlo runs to reduce realization dependency. In all experiments the output sequence is disturbed by additive white Gaussian noise for various SNR levels ranging from 7 to 27dB. The Normalized Mean Square Error, defined as  $\text{NMSE} = 10 \log_{10} \left( \mathbb{E}[\|\hat{\mathbf{h}} - \mathbf{h}\|_{\ell_2}^2] / \mathbb{E}[\|\mathbf{h}\|_{\ell_2}^2] \right)$ , was used to assess performance. The NMSE is computed after 500 iterations so that all algorithms have secured convergence.

A third order channel model was used to test the derived algorithms. The wireless channel taps for the linear and cubic part were generated by sparse Rayleigh fading rays. All rays are assumed to fade at the same Doppler frequency of  $f_D = 80\text{Hz}$  with sampling period  $T_s = 0.8\mu\text{s}$ . The linear and the cubic part have equal memory size  $M_1 = M_3 = 50$  and the support signal consists of 2 randomly selected elements for each part. The input signal is drawn from a complex Gaussian distribution  $\mathcal{CN}(0, 1/4)$ . We observe that the EM-Kalman and EM-RLS algorithms provide gains of 7dB and 5dB respectively, over the corresponding conventional non-sparse algorithms.

The choice of the parameters  $\gamma$ ,  $\lambda$  that were used to compare performance of the sparse algorithms are summarized in Table 2 for various SNR levels. The additional parameters required for the LMS are set to  $\mu = 5 \times 10^{-2}$  and  $\epsilon = 10$ . For

**Table 2.** Choice of parameters for the sparse algorithms

SNR	KF	RLS ( $\lambda, \gamma$ )	LMS ( $\ell_1, \log$ )
7	$3 \times 10^{-3}$	$.999, 1 \times 10^{-1}$	$5 \times 10^{-4}, 3 \times 10^{-3}$
11	$2 \times 10^{-3}$	$.999, 1 \times 10^{-1}$	$4 \times 10^{-4}, 2 \times 10^{-3}$
15	$1.4 \times 10^{-3}$	$.997, 9 \times 10^{-2}$	$3 \times 10^{-4}, 2 \times 10^{-3}$
19	$9 \times 10^{-4}$	$.995, 9 \times 10^{-2}$	$3 \times 10^{-4}, 1.5 \times 10^{-3}$
23	$6 \times 10^{-4}$	$.995, 8 \times 10^{-2}$	$3 \times 10^{-4}, 1.5 \times 10^{-3}$
27	$3.5 \times 10^{-4}$	$.991, 7 \times 10^{-2}$	$3 \times 10^{-4}, 1.5 \times 10^{-3}$

the EM-Kalman and the EM-RLS, the initial noise variance is set to  $\sigma_0^2 = 0.01\sigma_x^2$ .

It must be noted that due to the nature of the soft thresholding step, the identified  $\hat{\mathbf{h}}_n$  has many zero entries. This will allow to implement the EM-Kalman and EM-RLS algorithms in a low-complexity fashion similar to the approach taken in [4]. Thus, the EM-Kalman and EM-RLS algorithms introduce complexity gains as well as NMSE performance gains over the non-sparse methods.

#### V. CONCLUSIONS

In this paper, sparse approximations have been studied for nonlinear channel estimation. Adaptive algorithms combining Expectation-Maximization and Kalman filtering were developed and tested by simulations. Significant performance gains were achieved in comparison to the conventional non-sparse methods.

#### VI. REFERENCES

- [1] W.U. Bajwa, J. Haupt, G. Raz, and R. Nowak, "Compressed channel sensing," in *Proc. IEEE CISS*, 2008, pp. 5–10.
- [2] H. Ku and J.S. Kenney, "Behavioral modeling of nonlinear rf power amplifiers considering memory effects," *IEEE Trans. Microw. Theory Tech.*, vol. 51, no. 12, pp. 2495–2504, Dec. 2003.
- [3] Y. Chen, Y. Gu, and A. Hero, "Sparse LMS for system identification," in *Proc. IEEE ICASSP*, 2009.
- [4] B. Babadi, N. Kalouptsidis, and V. Tarokh, "Comparison of spaRLS and RLS algorithms for adaptive filtering," in *Proc. IEEE Sarnoff Symposium*, 2009.
- [5] S. Benedetto and E. Biglieri, *Principles of Digital Transmission: with wireless applications*, Springer, 1999.
- [6] M. Isaksson, D. Wisell, and D. Ronnow, "A comparative analysis of behavioral models for RF power amplifiers," *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 1, pp. 348–359, Jan. 2006.
- [7] J.A. Tropp, "Just relax: Convex programming methods for identifying sparse signals," *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 1030–1051, Mar. 2006.
- [8] L. Ljung, *General structure of adaptive algorithms: Adaptation and tracking*, in *Adaptive System Identification and Signal Processing Algorithms*, Eds. N. Kalouptsidis and S. Theodoridis, 1993.
- [9] M. Feder, *Statistical Signal Processing Using a Class of Iterative Estimation Algorithms*, Ph.D. thesis, M.I.T., Cambridge, MA, 1987.
- [10] A.H. Jazwinski, "Adaptive filtering," *Automatica*, vol. 5, pp. 475–485, 1969.